

WHAT IS CLAIMED IS:

- 1 1. A method for obtaining software performance data, said method comprising:
2 entering a software application;
3 setting a trace data flag to off;
4 while said software application has not exited, iteratively performing the following
5 steps for each module initiated by said software application:
6 setting said trace data flag to on if said module is registered with a performance
7 analyzer tool;
8 determining if said module includes trace data hooks;
9 if said module includes said trace data hooks and said trace data flag is on:
10 transmitting a request to said performance analyzer tool to record trace data in
11 response to encountering an embedded trace data hook in said module; and
12 if said module does not include said trace data hooks and said trace data flag is on:
13 transmitting a request to said performance analyzer tool to record trace data in
14 response to entry and exit of said module; and
15 receiving from said performance analyzer tool a report based on said trace data.

- 1 2. The method of claim 1 wherein said software application is written in the C
2 programming language.

- 1 3. The method of claim 1 wherein said software application is written in the C++
2 programming language.

- 1 4. The method of claim 1 wherein said software application is written in the
2 assembler programming language.

1 5. The method of claim 1 wherein said trace data in response to entry of said
2 module includes module name and time of module entry.

1 6. The method of claim 1 wherein said trace data in response to exit of said
2 module includes module name and time of module exit.

1 7. The method of claim 1 wherein said trace data hooks include:
2 program entry; and
3 program exit.

1 8. The method of claim 1 wherein said trace data hooks include:
2 function entry; and
3 function exit.

1 9. The method of claim 1 wherein said method for obtaining software performance
2 data further comprises:
3 transmitting a request to said performance analyzer tool for said report based on
4 said trace data.

1 10. The method of claim 1 wherein said report based on said trace data is in
2 graphical format.

1 11. The method of claim 1 wherein determining if said module includes trace data
2 hooks is performed by an operating system service.

1 12. The method of claim 1 wherein an operating system service determines if said
2 module is registered with said performance analyzer tool.

1 13. The method of claim 1 wherein said transmitting a request to said performance
2 analyzer tool to record trace data in response to entry and exit of said module is initiated
3 by an operating system service.

1 14. A system for obtaining software performance data, said system comprising a
2 computer processor, said computer processor implementing a method comprising:
3 entering a software application;
4 setting a trace data flag to off;
5 while said software application has not exited, iteratively performing the following
6 steps for each module initiated by said software application:

7 setting said trace data flag to on if said module is registered with a performance
8 analyzer tool;
9 determining if said module includes trace data hooks;
10 if said module includes said trace data hooks and said trace data flag is on:
11 transmitting a request to said performance analyzer tool to record trace data in
12 response to encountering an embedded trace data hook in said module; and
13 if said module does not include said trace data hooks and said trace data flag is on:
14 transmitting a request to said performance analyzer tool to record trace data in
15 response to entry and exit of said module; and
16 receiving from said performance analyzer tool a report based on said trace data.

1 15. The system of claim 14 wherein said software application is written in the C
2 programming language.

1 16. The system of claim 14 wherein said software application is written in the C++
2 programming language.

1 17. The system of claim 14 wherein said software application is written in the
2 assembler programming language.

1 18. The system of claim 14 wherein said trace data in response to entry of said
2 module includes module name and time of module entry.

1 19. The system of claim 14 wherein said trace data in response to exit of said
2 module includes module name and time of module exit.

1 20. The system of claim 14 wherein said embedded trace data hooks include:
2 program entry; and
3 program exit.

1 21. The system of claim 14 wherein said embedded trace data hooks include:
2 function entry; and
3 function exit.

1 22. The system of claim 14 wherein said method for obtaining software
2 performance data further comprises:
3 transmitting a request to said performance analyzer tool for said report based on
4 said trace data.

1 23. The system of claim 14 wherein said report based on said trace data is in
2 graphical format.

1 24. The system of claim 14 wherein determining if said module includes trace data
2 hooks is performed by an operating system service.

1 25. The system of claim 14 wherein an operating system service determines if said
2 module is registered with said performance analyzer tool.

1 26. The system of claim 14 wherein said transmitting a request to said
2 performance analyzer tool to record trace data in response to entry and exit of said module
3 is initiated by an operating system service.

1 27. A storage medium encoded with machine-readable computer program code for
2 obtaining software performance data, the storage medium storing instructions for causing a
3 software performance data system to implement a method comprising:

4 entering a software application;
5 setting a trace data flag to off;
6 while said software application has not exited, iteratively performing the following
7 steps for each module initiated by said software application:
8 setting said trace data flag to on if said module is registered with a performance
9 analyzer tool;
10 determining if said module includes trace data hooks;
11 if said module includes said trace data hooks and said trace data flag is on:
12 transmitting a request to said performance analyzer tool to record trace data in
13 response to encountering an embedded trace data hook in said module; and
14 if said module does not include said trace data hooks and said trace data flag is on:
15 transmitting a request to said performance analyzer tool to record trace data in
16 response to entry and exit of said module; and
17 receiving from said performance analyzer tool a report based on said trace data.

1 28. The storage medium of claim 27 wherein said software application is written
2 in the C programming language.

1 29. The storage medium of claim 27 wherein said software application is written
2 in the C++ programming language.

1 30. The storage medium of claim 27 wherein said software application is written
2 in the assembler programming language.

1 31. The storage medium of claim 27 wherein said trace data in response to entry of
2 said module includes module name and time of module entry.

1 32. The storage medium of claim 27 wherein said trace data in response to exit of
2 said module includes module name and time of module exit.

1 33. The storage medium of claim 27 wherein said trace data hooks include:
2 program entry; and
3 program exit.

1 34. The storage medium of claim 27 wherein said trace data hooks include:
2 function entry; and
3 function exit.

1 35. The storage medium of claim 27 further comprising instructions for causing
2 the software performance data system to implement:
3 transmitting a request to said performance analyzer tool for said report based on
4 said trace data.

1 36. The storage medium of claim 27 wherein said report based on said trace data is
2 in graphical format.

1 37. The storage medium of claim 27 wherein determining if said module includes
2 trace data hooks is performed by an operating system service.

1 38. The storage medium of claim 27 wherein an operating system service
2 determines if said module is registered with said performance analyzer tool.

1 39. The storage medium of claim 27 wherein said transmitting a request to said
2 performance analyzer tool to record trace data in response to entry and exit of said module
3 is initiated by an operating system service.